

# Algebraic Attack on the Alternating Step( $r, s$ ) Generator

Mehdi M. Hassanzadeh

The Selmer Center

Department of Informatics, University of Bergen

P.O. Box 7800, N-5020 Bergen, Norway

Email: Mehdi.hassanzadeh@ii.uib.no

Tor Helleseth

The Selmer Center

Department of Informatics, University of Bergen

P.O. Box 7800, N-5020 Bergen, Norway

Email: Tor.helleseth@ii.uib.no

**Abstract**—The Alternating Step( $r, s$ ) Generator, ASG( $r, s$ ), is a clock-controlled sequence generator which is recently proposed by A. Kanto. It consists of three registers of length  $l$ ,  $m$  and  $n$  bits. The first register controls the clocking of the two others. The two other registers are clocked  $r$  times (or not clocked) (resp.  $s$  times or not clocked) depending on the clock-control bit in the first register. The special case  $r = s = 1$  is the original and well known Alternating Step Generator. Kanto claims there is no efficient attack against the ASG( $r, s$ ) since  $r$  and  $s$  are kept secret. In this paper, we present an Alternating Step Generator, ASG, model for the ASG( $r, s$ ) and also we present a new and efficient algebraic attack on ASG( $r, s$ ) using  $3(m + n)$  bits of the output sequence to find the secret key with  $O((m^2 + n^2)2^{l+1} + m^3 2^{m-1} + n^3 2^{n-1})$  computational complexity. We show that this system is no more secure than the original ASG, in contrast to the claim of the ASG( $r, s$ )’s constructor.

## I. INTRODUCTION

The goal in stream cipher design is to efficiently produce pseudorandom sequences which should be indistinguishable from truly random sequences. From a cryptanalysis point of view, a good stream cipher should be resistant against a *known-plaintext attack*. In this kind of attack, the cryptanalyst is given a plaintext and the corresponding ciphertext, and the task is to determine the secret key. For a synchronous stream cipher, this is equivalent to the problem of finding the secret key or initial state that produced a given keystream output.

In stream cipher design, one usually uses Linear Feedback Shift Registers, LFSRs, as building block in different ways, and the secret key is often used as the initial state of the LFSRs. A general methodology for producing random-like sequences from LFSRs that has been popular is using the output of one or more LFSRs to control the clock of other LFSRs. The purpose is to destroy the linearity of the LFSR sequences and hence provide the resulting sequence with a large linear complexity. This structure is called a *Clock-Controlled Generator* which has several different types, e.g., Stop/Go Generator [2], [3], Step1/Step2 Generator [3], Shrinking Generator [4], Self-Shrinking Generator [5], and Jump Register which is proposed recently in [6]–[8] and it is used

in some candidates to the European ECRYPT/eSTREAM [9] project, e.g., Pomaranch [10] and Mickey [11].

An Alternating Step Generator (ASG), a well-known stream cipher proposed in [12], consists of a regularly clocked binary LFSR, **A**, and two Stop/Go clocked binary LFSRs, **B** and **C**. At each time, the clock-control bit from **A** determines which one of the two Stop/Go LFSRs is clocked, and the output sequence is obtained as bit-wise sum of the two Stop/Go clocked LFSR sequences.

ASG( $r, s$ ) proposed in [1] is a general form of an original ASG which will be described in the next section. The difference is that **B** and **C** are shifted  $r$  and  $s$  times, respectively, where  $r$  and  $s$  are part of the secret key. As far as we know, there is presently no efficient general attack on this algorithm. In this paper, we propose an algebraic attack on this algorithm and we will show that its security is no more than the security of the original ASG, in contrast to the constructor’s claim.

In Section II, a brief description of the ASG( $r, s$ ) will be presented and in Section III, the security of the ASG( $r, s$ ) is investigated from the author’s point of view. We model the ASG( $r, s$ ) to an original ASG in Section IV and according to this model, we will present our attack in Section V and conclude in Section VI.

## II. DESCRIPTION OF THE ASG( $r, s$ )

The Alternating Step( $r, s$ ) Generator, ASG( $r, s$ ), is a clock-controlled based stream cipher and it is similar to the original ASG but the clock-controlled LFSR **B** and **C** jump  $r$  and  $s$  steps respectively instead of in a Stop/Go manner.

ASG( $r, s$ ) is composed of a regularly clocked FSR, **A**, and two clock-controlled FSR’s, **B** and **C**. At each time, the clock-control bit from **A**, e.g.,  $0^{th}$  cell, determines which of the two FSR’s is clocked. **B** is clocked by the constant integer  $r$  and **C** is not clocked if the content of the  $0^{th}$  cell of **A** is ‘1’, otherwise, **B** is not clocked and **C** is clocked by the constant integer  $s$ . FSR **A** is called the Control Register and FSRs **B** and **C** are called the Generating Registers. The output bits of the ASG( $r, s$ ) are produced by adding modulo 2 the output bits of FSRs **B** and **C** under the control of FSR **A**. Kanto has recommended using a FSR **A** with a de-Bruijn output sequence of span  $l$  [14] and Primitive Linear Feedback Shift Register (LFSR) for generating registers **B** and **C** with length  $m$  and  $n$

This work was supported by the Norwegian Research Council and partially by the grant NIL-I-004 from Iceland, Liechtenstein and Norway through the EEA and Norwegian Financial Mechanisms.

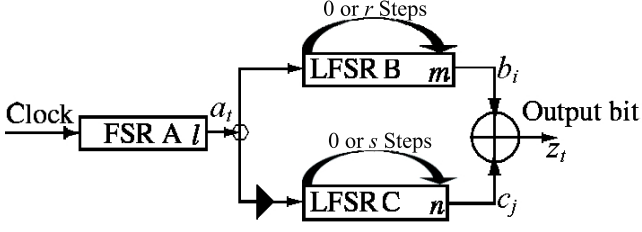


Fig. 1. The Alternating Step( $r, s$ ) Generator, ASG( $r, s$ )

bits respectively which is illustrated in fig. 1. He proved that when the values of  $m$  and  $n$  are satisfying  $\gcd(m, n) = 1$ , and the values of  $r$  and  $s$  are satisfying  $\gcd(r, 2^m - 1) = 1$  and  $\gcd(s, 2^n - 1) = 1$ , then the period of the output sequences is equal to  $2^l(2^m - 1)(2^n - 1)$  and the linear complexity ( $L_l$ ) of the output sequence satisfies  $(m + n)2^{l-1} < L_l \leq (m + n)2^l$ . The initial states of registers and the number of jumps,  $r$  and  $s$ , are the secret key. This structure is considered in the whole paper and in our attack.

### III. SECURITY OF THE ASG( $r, s$ )

Kanso claims in [1] that his structure, ASG( $r, s$ ), is secure against all known attacks so far. The output sequence of the ASG( $r, s$ ) is the XOR of its two irregularly decimated generating sequences. Thus, he claims one could not expect a strong correlation to be obtained efficiently, especially, if the primitive feedback polynomials of high Hamming weight are associated with the feedback functions of the generating registers **B** and **C** [23]. Furthermore, the values of  $r$  and  $s$  are considered as part of the secret key. Then, ASG( $r, s$ ) appears to be secure against all correlation attacks introduced in [20], [23]–[31].

Kanso also made the claim that ASG( $r, s$ ) is secure against algebraic attacks [13] and the complexity of this attack is equal to  $O((m^3 + n^3)\Phi 2^l)^1$  where  $\Phi = \Phi_1 \Phi_2$ ,  $\Phi_1$  is the number of possible values for  $r$  such that  $\gcd(r, 2^m - 1) = 1$  and  $\Phi_2$  is the number of possible values for  $s$  such that  $\gcd(s, 2^n - 1) = 1$ . This attack takes approximately  $O((m^3 + n^3)2^{m+n+l-2})$  steps using the estimate  $\Phi_1 = 2^{m-1}$  and  $\Phi_2 = 2^{n-1}$ . Therefore, the ASG( $r, s$ ) appears to be secure against this attack.

### IV. ASG MODEL FOR THE ASG( $r, s$ )

Throughout the paper, we refer to the output sequence of registers **A**, **B** and **C** by  $a = a_0, a_1, \dots, a_t$ ,  $b = b_0, b_1, \dots, b_i$  and  $c = c_0, c_1, \dots, c_j$  respectively. Furthermore, we refer to the output sequence of the ASG( $r, s$ ) by  $z = z_0, z_1, \dots, z_t$ . Let  $S_a(t)$ ,  $S_b(t)$  and  $S_c(t)$  denote the internal states of registers **A**, **B** and **C** at time  $t$  respectively, and let  $S_a(0)$ ,  $S_b(0)$  and  $S_c(0)$  denote their initial states. As the finite state machine is linear, the state transition can be described by a matrix which is the companion matrix for an LFSR. We refer to the transition matrix of registers **B** and **C** by  $T_b$  and  $T_c$  and we suppose that

<sup>1</sup>In [1], it is mentioned that this complexity is  $O(\Phi 2^l m^3 n^3)$  which is not correct.

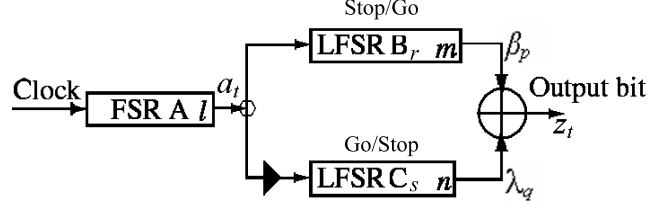


Fig. 2. ASG model for the ASG( $r, s$ )

the matrixes  $T_b$  and  $T_c$  are known in the rest of the paper. So, we have:

$$S_b(t) = S_b(t-1)T_b = S_b(0)T_b^t, \quad (1)$$

$$S_c(t) = S_c(t-1)T_c = S_c(0)T_c^t. \quad (2)$$

Suppose that  $z_t = b_i \oplus c_j$ , so we have:

$$z_{t+1} = (b_{i+r} \oplus c_j)a_t \oplus (b_i \oplus c_{j+s})(a_t \oplus 1). \quad (3)$$

Suppose that the first output bits of registers **B** and **C** are denoted by  $b_0$  and  $c_0$ . It is clear that only the bits in positions  $i = pr$  and  $j = qs$  are chosen from the regular output sequence of registers **B** and **C** respectively and other bits are discarded. In other words, the keystream output sequence ( $z_t$ ) is constructed by a combination of two  $r$ -decimated and  $s$ -decimated sequences derived from the regular output sequence of **B** and **C**. We refer to these irregular sequences by  $\beta$  and  $\lambda$  respectively. So, we have;  $\beta = \beta_0, \beta_1, \dots, \beta_t = b_0, b_{1r}, \dots, b_{tr}$ , such that  $\beta_t = b_{tr}$ , for all  $t \geq 0$  and  $\lambda = \lambda_0, \lambda_1, \dots, \lambda_t = c_0, c_{1s}, \dots, c_{ts}$ , such that  $\lambda_t = c_{ts}$ , for all  $t \geq 0$ .

The constructor Kanso [1] recommended that each register **B** and **C** should be an LFSR with output being an  $m$ -sequence. According to the following well known theorem from [14], both  $\beta$  and  $\lambda$  are  $m$ -sequences as well.

**Theorem 1:** Let  $b$  be a binary maximum-length sequence ( $m$ -sequence) with periodicity  $(2^m - 1)$ . Let  $\beta$  be a sequence obtained by sampling every  $r^{th}$  bit of  $b$ , starting with the first bit of  $b$ . Then  $\beta$  is again a  $m$ -sequence with the same period, if and only if  $\gcd(r, 2^m - 1) = 1$ .

This means that we can model the clock-controlled LFSRs **B** and **C**, by new regular LFSRs, **B<sub>r</sub>** and **C<sub>s</sub>**, with transition matrixes  $T_\beta$  and  $T_\lambda$  and regular output sequences  $\beta = \beta_0, \beta_1, \dots, \beta_t$  and  $\lambda = \lambda_0, \lambda_1, \dots, \lambda_t$  respectively. In other words, the sequences  $\beta$  and  $\lambda$  can be regenerated by the same length registers but different feedback polynomials. For their internal states, we have:

$$S_\beta(t) = S_\beta(t-1)T_\beta = S_\beta(0)T_\beta^t, \quad (4)$$

$$S_\lambda(t) = S_\lambda(t-1)T_\lambda = S_\lambda(0)T_\lambda^t. \quad (5)$$

If  $E_b$  and  $E_c$  denote the vectors which choose the last bit of registers **B** and **C**'s internal states as an output bit, we can write that:

$$\beta_t = S_\beta(t)E_b = S_\beta(0)T_\beta^t E_b, \quad (6)$$

$$\lambda_t = S_\lambda(t)E_c = S_\lambda(0)T_\lambda^t E_c. \quad (7)$$

Suppose that  $i = pr$  and  $j = qs$ , so we can rewrite (3) as follow:

$$z_t = b_i \oplus c_j = \beta_p \oplus \lambda_q, \\ z_{t+1} = (\beta_{p+1} \oplus \lambda_q) a_t \oplus (\beta_p \oplus \lambda_{q+1})(a_t \oplus 1). \quad (8)$$

It can be recognized easily that (8) describes an original ASG whose output ( $z_t$ ) is composed of  $\beta$  and  $\lambda$  under the control of  $a_t$ . So, we can model the ASG( $r, s$ ) to the original ASG described in (8) which is illustrated in fig. 2. In the next section, we will use this model and algebraic techniques to attack the ASG( $r, s$ ).

Several attacks have been proposed on the original ASG in the literature, but most of them do not affect the security of the ASG( $r, s$ ). Our idea can be applied to the original ASG, but it is not better than the previous attacks in contrast to the ASG( $r, s$ ).

Table I shows the complexity of the previous attacks and our attack on the original ASG. In table I, the first column shows the name of the previous attacks against the original ASG and the second column shows the **Minimum Keystream Length Requirement** (MKLR). The third column shows the total complexity and the last column shows the complexity of the attack in the case when  $l = m = n = 64$ . In table I and table II,  $L$  and  $M$  is equal to  $(l + m + n)$  and  $\max\{m, n\}$  respectively, and also we have  $\Gamma = 1 - 1/(0.19m + 3.1)$ .

We can see easily from table I that the Johansson's reduced complexity attack [20] is the best existing attack on the original ASG so far. For this reason, we briefly describe this attack and try to apply it to the ASG( $r, s$ ). In the Johansson's reduced complexity attacks, the adversary waits for a segment of  $M$  consecutive zeros (or ones) in the output sequence of the ASG. If  $m \leq n$ , then the adversary assumes that exactly  $M/2$  of them are from LFSR **B**. This is true with probability:

$$\left(\frac{M}{M/2}\right) 2^{-M}. \quad (9)$$

The remaining  $(m - M/2)$  bits of LFSR **B** are found by exhaustive search. The optimal complexity of this attack on the original ASG is  $O(m^2 2^{2m/3})$ .

This attack can not be applied to the ASG( $r, s$ ), because its main assumption, that exactly  $M/2$  bits of the  $M$ -bits output segment comes from LFSR **B**'s initial state, is only true when the output is composed of the two Stop/Go Generators' output. But in case of ASG( $r, s$ ), the values of  $r$  and  $s$  can be very large numbers. So, the main assumption to apply the Johansson's attack does not hold for the ASG( $r, s$ ) in general. Therefore, we have to apply this attack to our ASG model for the ASG( $r, s$ ), but it is not possible. Because the Johansson's attack needs to know the feedback polynomials of the generator registers, **B<sub>r</sub>** and **C<sub>s</sub>**, but they are unknown in our ASG model. So, we have to search the  $r$  and  $s$  values to apply this attack. We can search these values in  $\Phi$  steps and apply Johansson's attack for each value of the  $r$  and  $s$ . The optimal complexity of this attack is  $O(\Phi m^2 2^{2m/3}) = O(m^2 2^{8m/3})$ . In the next section, our attack on the ASG( $r, s$ ) will be explained and compared to this attack in table II.

TABLE I  
THE COMPLEXITY OF PREVIOUS ATTACKS AGAINST THE ORIGINAL ASG

Attack	MKLR	Complexity	$l = m = n = 64$
Edit Distance Correlation [16]–[18]	$O(m + n)$	$O((m + n)2^{m+n})$	$2^{135}$
Clock Control Guessing Attack [22]	$l + m + n$	$O(L^3 2^{L/2})$	$2^{118.8}$
Algebraic Attack [13]	$O(m + n)$	$O((m^3 + n^3)2^l)$	$2^{83}$
Edit Probability Correlation Attack [19]	$O(m + n)$	$O(M^2 2^M)$	$2^{76}$
Khazaei's Reduced Complexity Attack [21]	$2m$	$O(m^2 2^{\Gamma m})$	$2^{71.8}$
Improved Edit Distance Correlation [32]	$O(M)$	$O(M 2^M)$	$2^{70}$
Linear Consistency Attack [15]		$O(\min(m, n)2^l)$	$2^{70}$
Johansson's Reduced Complexity Attacks [20]	$O(2^{2m/3})$	$O(m^2 2^{2m/3})$	$2^{54.7}$
Our Algebraic Attack	$3(m + n)$	$O((m^2 + n^2)2^{l+1})$	$2^{78}$

## V. OUR ALGEBRAIC ATTACK ON THE ASG( $r, s$ )

The goal of an attack on the stream cipher is to recover the secret key or to predict and reproduce the rest of the keystream to recover the rest of the cipher text. In [13] an algebraic attack approach to a family of irregularly clock-controlled LFSR based systems is presented. The complexity of this attack on the original ASG structure is  $O((m^3 + n^3)2^l)$ . But, its complexity on the ASG( $r, s$ ) is approximately  $O((m^3 + n^3)2^{l+m+n-2})$ . We make use of the same idea to attack the ASG( $r, s$ ) but we have improved its complexity significantly. If we XOR  $z_t$  by  $z_{t+1}$  from (8), we have:

$$z_t \oplus z_{t+1} = \beta_p \oplus \lambda_q \oplus (\beta_{p+1} \oplus \lambda_q) a_t \oplus (\beta_p \oplus \lambda_{q+1})(1 \oplus a_t). \quad (10)$$

Now, if we multiply both sides of (10) by  $a_t$ , we have:

$$(z_t \oplus z_{t+1})(a_t) = (\beta_p \oplus \beta_{p+1})(a_t), \quad (11)$$

and if we multiply both sides of (10) by  $(1 \oplus a_t)$ , we obtain:

$$(z_t \oplus z_{t+1})(1 \oplus a_t) = (\lambda_q \oplus \lambda_{q+1})(1 \oplus a_t). \quad (12)$$

From (11) and (12) we conclude that:

$$if \quad a_t = \begin{cases} 1 & \beta_{p+1} = \beta_p \oplus z_t \oplus z_{t+1} \\ 0 & \lambda_{q+1} = \lambda_q \oplus z_t \oplus z_{t+1} \end{cases}. \quad (13)$$

So, if we know the value of  $a_t$ ,  $\beta_p$  and  $\lambda_q$ , we can find  $\beta_{p+1}$  and  $\lambda_{q+1}$ . Note that  $z_t$  and  $z_{t+1}$  belong to the known output sequence of the ASG( $r, s$ ).

In our attack, we search over all possible values for the initial state of register **A** and produce the sequence  $a =$

$a_0, a_1, \dots, a_t$ . Then, we guess the value of  $\beta_0$  and calculate  $\lambda_0 = z_0 \oplus \beta_0$ . Now, by (13) we can find the bits  $\beta_p$  for  $p \geq 1$  and  $\lambda_q$  for  $q \geq 1$  as much as needed.

Using the Berlekamp-Massey algorithm and  $2m$  bits of  $\beta$  and  $2n$  bits of  $\lambda$ , we can find the feedback polynomials and the initial states of the generator registers,  $\mathbf{B}_r$  and  $\mathbf{C}_s$ , that can directly produce the sequences  $\beta$  and  $\lambda$  regularly. Then, by the rest of the output sequence we can test our guesses for the value of  $\beta_0$  and the initial state of register  $\mathbf{A}$ . As the complexity of Berlekamp-Massey algorithm is  $O(n^2)$  for a sequence of length  $n$ , the complexity of this part of our attack is equal to  $O((m^2 + n^2)2^{l+1})$ .

Now, we have to find the value of parameters  $r$  and  $s$  and the initial states of LFSR  $\mathbf{B}$ ,  $S_b(0)$ , and  $\mathbf{C}$ ,  $S_c(0)$ . We first have to represent  $b_{rt}$  and  $b_t$  by the *Trace Function*. The trace function,  $Tr_m(x)$ , is a mapping from the finite field  $GF(2^m)$  to  $GF(2)$  defined by

$$Tr_m(x) = \sum_{i=0}^{m-1} x^{2^i}.$$

Any  $m$ -sequence  $\{b_t\}$  of period  $2^m - 1$  with characteristic polynomial which is the minimal polynomial of a primitive element  $\alpha$  (of order  $2^m - 1$ ) in  $GF(2^m)$  can be represented by the trace function as  $b_t = Tr_m(u\alpha^t)$ . Every nonzero element  $u \in GF(2^m)$  corresponds to a cyclic shift of  $\{b_t\}$ . In our case the situation is that we know  $T_b$  and have found  $T_\beta$  and we want to find  $r$  and  $s$ . To find  $r$  we know already  $\alpha$  and  $b_{rt}$  for  $\{t = 0, 1, 2, \dots\}$  as well as the relations (14) and (15).

$$b_t = Tr_m(u\alpha^t), \quad (14)$$

$$b_{rt} = Tr_m(u\alpha^{rt}) = Tr_m(u\gamma^t), \quad (15)$$

where  $(\gamma = \alpha^r)$ . We want to find  $u$  which is part of the key since it determines  $\{b_t\}$ . First we guess a possible value for  $r$  and compute  $\gamma = \alpha^r$ . Then we construct an equation system by (15) for  $\{t = 0, 1, 2, \dots, m-1\}$ . This is an equation system in the  $m$  unknowns  $u, u^2, \dots, u^{2^{m-1}}$ . The system has full rank due to the special form of the coefficient matrix and can therefore be solved in complexity  $O(m^3)$ . If the solution of equation system,  $u$ , can regenerate correctly the sequence  $b_{rt}$  by using (15) for  $\{t = m, m+1, \dots\}$  for sufficiently many bits, our guess for  $r$  is correct. In other case, we have to repeat this process with new possible value for  $r$ . Then  $u$  is found and we can generate  $b_t$  by using (14) for  $\{t = 0, 1, 2, \dots, m-1\}$  which is the initial state of LFSR  $\mathbf{B}$ . Similarly we can find the initial state in the other LFSR  $\mathbf{C}$ . The complexity of this part is equal to  $O(\Phi_1 m^3 + \Phi_2 n^3) = O(m^3 2^{m-1} + n^3 2^{n-1})$ . Therefore, the total complexity of our attack is equal to:

$$C = O((m^2 + n^2)2^{l+1} + m^3 2^{m-1} + n^3 2^{n-1}). \quad (16)$$

Table II shows the complexity of the previous attacks and our attack on the ASG( $r, s$ ) to compare their efficiencies. In case of  $l = m = n = 64$ , the best previous attack needs  $2^{153.5}$  steps to break the ASG( $r, s$ ), but our attack is significantly better and it can find the secret key only by  $2^{82}$  steps. This

TABLE II  
COMPARISON OF OUR ATTACK ON THE ASG( $r, s$ ) WITH OTHER KNOWN ATTACKS

Attack	MKLR	Complexity	$l = m = n = 64$
Clock Control Guessing Attack [22]	$l + m + n$	$O(L^3 2^{\frac{L+2m+2n-4}{2}})$	$2^{566}$
Edit Distance Correlation [16]–[18]	$O(m + n)$	$O((m + n)2^{2(m+n)-2})$	$2^{261}$
Algebraic Attack [13]	$O(m + n)$	$O((m^3 + n^3)2^{L-2})$	$2^{209}$
Edit Probability Correlation Attack [19]	$O(m + n)$	$O(M^2 2^{M+m+n-2})$	$2^{202}$
Improved Edit Distance Correlation [32]	$O(M)$	$O(M 2^{M+m+n-2})$	$2^{196}$
Linear Consistency Attack [15]		$O(\min(m, n)2^{3l-2})$	$2^{196}$
Khazaei's Reduced Complexity Attack [21]	$2m$	$O(m^2 2^{(\Gamma+2)(m-2)})$	$2^{167.5}$
Johansson's Reduced Complexity Attacks [20]	$O(2^{2m/3})$	$O(m^2 2^{(8m/3)-2})$	$2^{153.5}$
Our Algebraic Attack	$3(m + n)$	$O((m^2 + n^2)2^{l+1} + m^3 2^{m-1} + n^3 2^{n-1})$	$2^{82}$

difference comes from our idea to find the values of  $r$  and  $s$ . In the previous attacks, the adversary has to guess the values of  $r$  and  $s$  by exhaustive search, and for each guess, the attack must be applied to the algorithm. But, in our idea, we do not need to know the values of  $r$  and  $s$  to apply our attack and we find these values independent of the exhaustive search over the initial state of register  $\mathbf{A}$ .

## VI. CONCLUSION

In this paper, we present an ASG model for the ASG( $r, s$ ) and also we present a new algebraic attack against the ASG( $r, s$ ). The designer of the ASG( $r, s$ ) claims that this structure is more secure than the original ASG, but we show that its security is not more than the original ASG. Our attack can find the secret key of ASG( $r, s$ ) by using of  $3(m + n)$  bits of the output keystream with  $O((m^2 + n^2)2^{l+1} + m^3 2^{m-1} + n^3 2^{n-1})$  computational complexity.

As far as we know, there is no efficient attack against the ASG( $r, s$ ) so far. The complexity of previous attacks is much higher than the complexity of our attack. In case of  $l = m = n = 64$ , the best previous attack needs  $2^{153.5}$  steps to break the ASG( $r, s$ ), but our attack can find the secret key only by  $2^{82}$  steps. Our attack can be applied to the original ASG structure. Its complexity is comparable to the best known attacks but our attack does not need to know the characteristic polynomial of generating registers. Applying our idea to other clock-controlled structures is a subject for future research.

## REFERENCES

- [1] A. Kansa, "The Alternating Step( $r, s$ ) Generator", SECI02, Tunis, Sep. 2002.
- [2] T. Beth and F. Piper, "The Stop and Go Generator", *Advances in Cryptology: Proceedings of Eurocrypt'84*, LNCS, Berlin: Springer-Verlag, vol. 209, pp. 88-92, 1985.
- [3] D. Gollmann and W. Chambers, "Clock-Controlled Shift Register: A Review", *IEEE J. Selected Areas Communications*, vol. 7, NO. 4, pp. 525-533, May 1989.
- [4] D. Coppersmith, H. Krawczyk, and Y. Mansour, "The Shrinking Generator", *CRYPTO'93*, LNCS, vol. 773, pp. 22-39, Springer, Berlin, 1993.
- [5] W. Meier and O. Staffelbach, "The self-shrinking generator", In A. De Santis, editor, *Advances in Cryptology - Eurocrypt'94*, LNCS, vol. 950, pp. 205-214, Springer, Berlin, 1995.
- [6] C.J.A. Jansen: Modern stream cipher design: "A new view on multiple clocking and irreducible polynomials", *Actas de la VII Reunión Española sobre Criptología y Seguridad de la Información*, S. González, C. Martínez, Eds. Tomo I, pp. 11-29, Oviedo, 2002.
- [7] C.J.A. Jansen: Partitions of polynomials: "Stream ciphers based on jumping shift registers" Cardinal, J., Cerf, N., Delgrange, O., Markowitch, O. (eds.) 26th Symposium on Inf. Theory in the Benelux, Enschede, Werkgemeenschap voor Informatie- en Communicatietheorie, pp. 277-284, 2005.
- [8] C.J.A. Jansen: "Stream cipher design based on jumping finite state machines", *Cryptology ePrint Archive*, Report 2005/267, 2005. <http://eprint.iacr.org/2005/267/>.
- [9] Ecrypt Stream Cipher Project. <http://www.ecrypt.eu.org/stream/>
- [10] C.J.A. Jansen, T. Hellese, A. Kholosha, "Cascade Jump Controlled Sequence Generator and Pomaranch Stream Cipher", LNCS, volume 4986, pp. 224-243, Springer, 2008. [www.ecrypt.eu.org/stream](http://www.ecrypt.eu.org/stream)
- [11] S. Babbage and M.Dodd, "The MICKEY Stream Ciphers", LNCS, vol. 4986, pp. 191-209, Springer, and the ECRYPT/eSTREAM project: 224-243, 2008. <http://www.ecrypt.eu.org/stream/Micky.html>
- [12] C. G. Günther, "Alternating Step Generators Controlled by De Bruijn Sequences". *Advances in Cryptology: Eurocrypt 87*, LNCS, Spingler-Verlag, vol. 309, 1988, pp. 5-14.
- [13] S. Al-Hinai, L. Batten, B. Colbert, and K. Wong, "Algebraic Attacks on Clock-Controlled Stream Ciphers", LNCS, Volume 4058, pages 1-16, Springer Berlin, Heidelberg, 2006.
- [14] S. W. Golomb, "Shift register sequences", Holden-Day, Inc. San Francisco, CA, 1967, Revised second edition, Aegean Park Press, Laguna Hills, CA, 1982.
- [15] K. Zeng, C. H. Yang, and T. R. N. Rao, "On the Linear Consistency Test (LCT) in Cryptanalysis with Applications", In *CRYPTO'89*, pp. 164-174, 1989.
- [16] J. Golic and R. Menicocci, "Edit Distance Correlation Attack on the Alternating Step Generator", In *CRYPTO'97*, pp. 499-512, 1997.
- [17] J. Golic, "Embedding probabilities for the Alternating Step Generator", In *IEEE Transactions on Information Theory* 51(7), pp. 2543-2553, 2005.
- [18] S. Jiang and G. Gong, "On Edit Distance Attack to Alternating Step Generator", In *Other Combinatorial Structures*: pp. 85-92, 2003.
- [19] J. Dj. Golic and R. Menicocci, "Edit Probability Correlation Attack on the Alternating Step Generator", In *Sequences and Their Applications - SETA 1998*.
- [20] T. Johansson, "Reduced Complexity Correlation Attacks on Two Clock-Controlled Generators", In *ASIACRYPT'98*, pp. 342-356, 1998.
- [21] S. Khazaei, S. Fischer and W. Meier, "Reduced Complexity Attacks on the Alternating Step Generator", *Proceedings of Selected Areas in Cryptography, SAC'07*, LNCS, vol. 4876, pp. 1-16, 2007.
- [22] E. Zenner, "On the efficiency of the clock control guessing attack", *ICISC*, pp. 200-212, 2003.
- [23] W. Meier, O. Staffelbach, "Fast correlation attacks on certain stream ciphers", *Journal of Cryptology*, vol. 1, pp. 159-176, 1989.
- [24] J. Golic, M. Mihaljevic, "A generalized correlation attack on a class of stream ciphers based on the Levenstein distance", *Journal of Cryptology*, vol. 3, pp. 201-212, 1991.
- [25] J. Golic, "Towards fast correlation attacks on irregularly clocked shift registers", In *Advances in Cryptology: EuroCrypt'95*, LNCS, Springer-Verlag, vol. 921, pp. 248-262, 1995.
- [26] T. Siegenthaler, "Correlation-immunity of non-linear combining functions for cryptographic applications", *IEEE Transactions On Information Theory*, vol. IT-30, no. 5, pp.776-779, 1984.
- [27] J. Golic, "On the security of shift register based keystream generators", In R. Anderson, Editor, *FSE, Cambridge Security Workshop*, LNCS, Berlin: Springer-Verlag, vol. 809, pp. 90-100, 1994.
- [28] M. Mihaljevic, "An approach to the initial state reconstruction of a clock-controlled shift register based on a novel distance measure", *Advances in Cryptology: AusCrypt 92*, LNCS, vol. 178, pp. 349-356, 1993.
- [29] J. Golic, L. O'Connor, "Embedding probabilistic correlation attacks on clock-controlled shift registers", *Advances in Cryptology: EuroCrypt'94*, LNCS, vol. 950, pp. 230-243, 1995.
- [30] T. Johansson, F.Jonsson, "Improved fast correlation attacks on certain stream ciphers via convolutional codes", In *Advances in Cryptology: EuroCrypt'99*, LNCS, vol. 1592, Springer-Verlag, pp. 347-362, 1999.
- [31] T. Johansson, F.Jonsson, "Fast correlation attacks through reconstruction of linear polynomials", In *Advances in Cryptology: Crypto 2000*, LNCS, vol. 1880, Springer-Verlag, pp. 300-315, 2000.
- [32] J. Golic and R. Menicocci, "Correlation analysis of the alternating step generator", *Des. Codes Cryptography*, 31(1), pp. 51-74, 2004.